UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/753,279	12/30/2000	Gregory Hibdon	42390P9571	1479

7590 01/03/2005

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Seventh Floor
12400 Wilshire Boulevard
Los Angeles, CA 90025-1026

EXAMINER	
YIGDALL, MICHAEL J	

ART UNIT	PAPER NUMBER
2122	

DATE MAILED: 01/03/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/753,279	HIBDON, GREGORY	
	Examiner	Art Unit	
	Michael J. Yigdall	2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 October 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-3,5-12,14-18 and 20-54* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3,5-12,14-18 and 20-54* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on October 7, 2004 has been entered. Claims 1-3, 5-12, 14-18 and 20-54 are now pending.

Response to Arguments

2. Applicant's arguments with respect to claims 1-3, 5-12, 14-18 and 20-24 have been considered but are moot in view of the new ground(s) of rejection.

Claim Objections

3. Claim 38 is objected to because of the following informalities: The claim lacks antecedent basis for "the scan insertion module" and "repeating determining." The claim was perhaps intended to depend from claim 37, rather than from claim 33, and has been interpreted accordingly. Appropriate correction is required.

4. Claim 42 is objected to because of the following informalities: The claim does not end with a period, as per MPEP § 608.01(m), and the word "added" in line 3 should perhaps be replaced with --adding--. Appropriate correction is required.

5. Claim 43 objected to because of the following informalities: The phrase "is a symbol does not" in line 7 should perhaps be replaced with --is a symbol that does not--. Appropriate correction is required.

6. Applicant is respectfully asked for cooperation in finding and correcting all such informalities in the claims.

Claim Rejections - 35 USC § 101

7. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

8. Claims 27-32 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

With respect to claims 27-32, the claims recite a "token object" having variables, strings and other indicators, which is merely an arrangement of data *per se*. Such descriptive material that does not exhibit any functional interrelationship with the way in which computing processes are performed does not constitute a statutory process, machine, manufacture or composition of matter. See MPEP § 2106(IV)(B)(1)(b).

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are

such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 1-3, 10-12,16-18, 25 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,946,488 to Tanguay et al. (art of record, hereinafter "Tanguay") in view of U.S. Pat. No. 6,182,281 to Nackman et al. (art made of record, hereinafter "Nackman").

With respect to claim 1 (currently amended), Tanguay discloses a method comprising:

(a) reading a line of data from a file containing source code written in a high level language (see, for example, block 306 in FIG. 3 and column 5, lines 9-13, which shows reading lines from a source file, and column 4, line 65 to column 5, line 1, which shows that the source code is written in a high level language);

(b) generating a stream of tokens from said line of data (see, for example, block 308 in FIG. 3 and column 5, lines 14-16, which shows translating the source code into a stream of tokens), said stream of tokens representing any of a specific type of macro in the line of data as being expanded while other types of macros are not expanded (see, for example, column 1, lines 61-66, which shows selecting specific macros for expansion, such as based on the type of the macro, while others are not expanded).

Although Tanguay inherently represents and stores each token in some form so as to process the tokens, Tanguay does not expressly disclose:

(c) generating a token object for each token, the token object including a visibility variable to represent whether a parser and an output module may view the respective token.

However, Nackman discloses reading the source code of a program and generating tokens (see, for example, blocks 32 and 34 in FIG. 4), in a system for the incremental compilation of high-level languages (see, for example, the title). Such a system greatly reduces the compilation time during program development and maintenance (see, for example, column 2, lines 24-27). Nackman further discloses generating and persisting objects for the tokens represented in the program (see, for example, column 3, lines 17-24), such as macro objects (see, for example, column 7, lines 36-37). The macro objects include a hidden status, i.e. a visibility variable (see, for example, column 10, lines 31-39), to indicate whether other modules may view the macro (see, for example, column 10, lines 1-4).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the system of Tanguay with token objects that include visibility variables, such as the objects taught by Nackman, so as to persist the program representation and enable incremental compilation, thereby reducing the time to compile during program development and maintenance.

Tanguay also discloses:

(d) parsing the stream of tokens using a parser and with reference to respective token objects (see, for example, block 310 in FIG. 3 and column 5, lines 17-18, which shows parsing the stream of tokens to execute preprocessing directives and expand macros);

(e) inserting commands representing operations to be performed by a macro into said stream of tokens if a macro is present (see, for example, column 5, lines 61-62, which shows expanding a macro by inserting the macro definition, i.e. into the stream of tokens); and

(f) writing the stream of tokens to an output file using an output module and with reference to respective token objects (see, for example, column 4, lines 35-47, which shows writing code, i.e. the stream of tokens, to an output file).

With respect to claim 2 (currently amended), Tanguay also discloses the limitation wherein generating a stream of tokens further comprises:

(a) determining whether tokens are present in either an input file, a look-ahead buffer, or a macro expansion list (see, for example, column 8, lines 61-63, which shows reading new tokens from a source file; also see, for example, column 9, lines 13-21, which shows a string table serving as a look-ahead buffer, and column 9, lines 28-34, which shows a representation comprising macro expansion operators, i.e. a macro expansion list); and

(b) responsive to finding tokens, reading the tokens first from said look-ahead buffer, then from said macro expansion list, then from said input file (see, for example, column 9, lines 13-21, which shows that tokens are first identified from the string table serving as a look-ahead buffer, column 9, lines 39-45, which shows that the string table then identifies tokens in macro expansions, and column 8, lines 61-63, which shows that new tokens, i.e. tokens not yet identified, are read from the source file);

(c) presenting the tokens to a parser so that any macro in the line of data appears to have been expanded (see, for example, column 9, lines 53-59, which shows presenting the tokens, including the expanded code, to either a viewer or a compiler, i.e. a parser).

With respect to claim 3 (currently amended), Tanguay also discloses the limitation wherein parsing further comprises:

(a) reading a token (see, for example, column 8, lines 61-67, which shows reading tokens);

(b) determining a type of the read token (see, for example, column 8, lines 61-67, which shows determining the type of each new token);

(c) responsive to determining that the read token is an end-of-line, processing an input line of tokens (see, for example, column 8, lines 61-67, which shows identifying syntactic elements, such as end-of-line tokens, and column 5, lines 12-16, which further shows that the tokens are processed in terms of input lines);

(d) responsive to determining that the read token is not a symbol, adding the read token to a current line token list (see, for example, column 9, lines 7-12, which shows adding tokens to a table or list, and column 11, lines 10-16, which shows a line database for storing information related to lines, i.e. lines comprised of tokens);

(e) responsive to determining that the read token is a symbol that indicates a beginning of a macro definition, recording a macro name and the macro definition and adding the read token to a look-ahead buffer (see, for example, column 9, lines 28-34, which shows identifying the beginning of a macro expansion or definition, and column 9, lines 13-21, which shows adding tokens to a string table serving as a look-ahead buffer; also see, for example, column 11, lines 33-36, which shows a macro database having records of macro references and expansions, i.e. macro names and definitions); and

(f) responsive to determining that the read token is a symbol that does not indicate a beginning of a macro definition, adding the read token to a current line token list (see, for example, column 9, lines 7-12, which shows adding tokens to a table or list, and column 11, lines

10-16, which shows a line database for storing information related to lines, i.e. lines comprised of tokens).

With respect to claim 10 (currently amended), the limitations recited in the claim are analogous to those of claim 1 (see Tanguay and Nackman as applied to claim 1 above). Note that Tanguay also discloses:

(a) a storage device having stored therein one or more routines for selectively expanding macros within source code (see, for example, memory 130 in FIG. 1, which shows a storage device having a selective preprocessor; also see, for example, column 1, lines 61-66, which shows selectively expanding macros in source code); and

(b) a processor coupled to the storage device for executing the one or more routines for selectively expanding macros within source code (see, for example, CPU 170 and bus 180 in FIG. 1, which shows a processor coupled to the storage device).

With respect to claim 11 (currently amended), the limitations recited in the claim are analogous to those of claim 2 (see Tanguay and Nackman as applied to claim 2 above).

With respect to claim 12 (currently amended), the limitations recited in the claim are analogous to those of claim 3 (see Tanguay and Nackman as applied to claim 3 above).

With respect to claim 16 (currently amended), the limitations recited in the claim are analogous to those of claim 1 (see Tanguay and Nackman as applied to claim 1 above). Note that Tanguay also discloses a machine-readable medium (see, for example, memory 130 in FIG. 1) and a processor (see, for example, CPU 170 in FIG. 1).

With respect to claim 17 (currently amended), the limitations recited in the claim are analogous to those of claim 2 (see Tanguay and Nackman as applied to claim 2 above).

With respect to claim 18 (currently amended), the limitations recited in the claim are analogous to those of claim 3 (see Tanguay and Nackman as applied to claim 3 above).

With respect to claim 25 (new), Tanguay also discloses the limitation wherein writing comprises:

(a) writing expanded macro tokens to the output file if the macro is of the specific type of macro (see, for example, selective preprocessor 200 in FIG. 2 and column 4, lines 35-47, which shows writing code in expanded form, i.e. code including expanded macro tokens, to an output file; also see, for example, column 4, lines 48-62, which further shows expanding specific macros based on user input, such as according to the type of macro, and using the same selection function in a process such as compilation); and

(b) writing an original macro call to the output file if the macro is not the specific type of macro (see, for example, selective preprocessor 200 in FIG. 2 and column 4, lines 35-47, which shows writing original code, i.e. code including original macro calls, to an output file; also see, for example, column 4, lines 48-62, which further shows expanding specific macros based on user input, such as according to the type of macro, and using the same selection function in a process such as compilation).

With respect to claim 26 (new), the limitations recited in the claim are analogous to those of claim 2 (see Tanguay and Nackman as applied to claim 2 above).

11. Claims 5-9, 14, 15, 20-24 and 33-54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tanguay in view of Nackman, as applied to claims 1, 10 and 16 above, respectively, and further in view of U.S. Pat. No. 6,581,191 to Schubert et al. (art made of record, hereinafter "Schubert").

With respect to claim 5 (currently amended), although Tanguay discloses debugging source code (see, for example, column 2, lines 2-3) written in any language that supports preprocessing (see, for example, column 5, lines 37-42), which would encompass hardware description languages, Tanguay in view of Nackman does not expressly disclose the limitation wherein source code written in a high level language comprises a hardware description language (HDL) for representing hardware designs.

However, Schubert discloses debugging hardware designs that are written in hardware description languages (see, for example, the abstract).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use hardware description languages, such as taught by Schubert, in the system of Tanguay and Nackman, so as to debug hardware designs.

With respect to claim 6 (currently amended), although Tanguay discloses debugging source code (see, for example, column 2, lines 2-3) and expanding any macro (see, for example, column 2, lines 42-44), which would encompass scan macros, Tanguay in view of Nackman does not expressly disclose the limitation wherein specific type of macro comprises a scan macro.

However, Schubert discloses debugging hardware designs that are written in hardware description languages (see, for example, the abstract). Schubert further discloses instrumenting the hardware design (see, for example, column 13, line 48 to column 14, line 7) based on instrumentation directives, such as pragmas or macros (see, for example, column 23, lines 43-62). The instrumentation is analogous to scan insertion (see, for example, column 26, line 57 to column 27, line 2), and thus the instrumentation directives function as scan macros.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use scan macros, such as taught by Schubert, in the system of Tanguay and Nackman, so as to debug hardware designs.

With respect to claim 7 (currently amended), Tanguay discloses a method for debugging software (see, for example, column 2, lines 2-3) comprising:

(a) reading source code, the source code including a plurality of macro definitions (see, for example, column 8, lines 61-63, which shows reading source code, and column 1, lines 61-66, which further shows that the source code includes macro definitions);

(b) creating a token stream based on the source code that includes multifaceted tokens that can be hidden from or made visible to a subsequent parsing process by expanding the plurality of macro definitions and making tokens associated with some macros visible to the subsequent parsing process and marking other tokens as hidden (see, for example, column 5, lines 14-16, which shows translating the source code into a stream of tokens; also see, for example, column 4, lines 52-62, which shows expanding and contracting macro definitions to make the corresponding tokens visible and hidden, respectively, to a subsequent process such as compilation).

Although Tanguay inherently represents and stores each token in some form so as to process the tokens, Tanguay does not expressly disclose the limitation wherein the multifaceted tokens are associated with a token object for each token, the token object including a visibility variable to represent whether a parser and an output module may view the respective token.

However, Nackman discloses reading the source code of a program and generating tokens (see, for example, blocks 32 and 34 in FIG. 4), in a system for the incremental compilation of high-level languages (see, for example, the title). Such a system greatly reduces the compilation time during program development and maintenance (see, for example, column 2, lines 24-27). Nackman further discloses generating and persisting objects for the tokens represented in the program (see, for example, column 3, lines 17-24), such as macro objects (see, for example, column 7, lines 36-37). The macro objects include a hidden status, i.e. a visibility variable (see, for example, column 10, lines 31-39), to indicate whether other modules may view the macro (see, for example, column 10, lines 1-4).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the system of Tanguay with token objects that include visibility variables, such as the objects taught by Nackman, so as to persist the program representation and enable incremental compilation, thereby reducing the time to compile during program development and maintenance.

Tanguay also discloses:

(c) performing macro expansion using a parser by parsing those of the multifaceted tokens that are visible to the parser based on said visibility variable and adding appropriate commands (see, for example, column 5, lines 17-18, which shows parsing the stream of tokens to

execute preprocessing directives and expand macros, and column 5, lines 61-62, which shows expanding a macro by adding the macro definition, i.e. to the token stream); and

(d) using an output module to generate an expanded source code file containing expanded versions of the macro definitions which are visible to the output module based on the visibility variable and which are selected but that omits expanded versions of those that are not selected (see, for example, column 4, lines 35-47, which shows a selective preprocessor for generating an expanded source code file having the original code or the original code with expanded macro definitions).

Tanguay in view of Nackman does not disclose expressly the limitations wherein:

(i) the source code is a hardware description language (HDL) representation of a hardware design;

(ii) some of the macros relate to scan insertion;

(iii) the token object includes a scan variable to represent whether the respective token is related to scan;

(iv) scan commands are added to the representation; and

(v) the output file is a scan inserted HDL file.

However, Schubert discloses debugging hardware designs that are written in hardware description languages (see, for example, the abstract). Schubert further discloses instrumenting the hardware design and outputting an instrumented HDL file (see, for example, column 13, line 48 to column 14, line 7) based on instrumentation directives, such as pragmas or macros (see, for example, column 23, lines 43-62). The instrumentation is analogous to scan insertion (see, for example, column 26, line 57 to column 27, line 2), and thus the instrumentation directives

function as scan macros. Schubert further discloses comment objects that include indicators to represent whether the comment is an instrumentation directive related to scan, so as to differentiate such instrumentation directives from other comments (see, for example, column 23, line 63 to column 24, line 4).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the system of Tanguay and Nackman with the features taught by Schubert, so as to debug hardware designs. Furthermore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the token objects of Tanguay and Nackman with a scan variable, such as the indicator taught by Schubert, so as to differentiate the tokens that are related to scan from other tokens.

With respect to claim 8 (currently amended), Schubert further discloses the limitation wherein HDL comprises a high-level language (see, for example, column 8, lines 53-63).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use hardware description languages comprising high-level languages, such as taught by Schubert, in the system of Tanguay and Nackman, so as to debug hardware designs.

With respect to claim 9 (currently amended), Schubert further discloses the limitation wherein said hardware design represents an integrated circuit design (see, for example, column 8, lines 16-26).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use integrated circuit designs, such as taught by Schubert, in the system of Tanguay and Nackman, so as to debug hardware designs.

With respect to claim 14 (currently amended), the limitations recited in the claim are analogous to those of claim 5 (see Tanguay, Nackman and Schubert as applied to claim 5 above).

With respect to claim 15 (currently amended), the limitations recited in the claim are analogous to those of claim 6 (see Tanguay, Nackman and Schubert as applied to claim 6 above).

With respect to claim 20 (currently amended), the limitations recited in the claim are analogous to those of claim 5 (see Tanguay, Nackman and Schubert as applied to claim 5 above).

With respect to claim 21 (currently amended), the limitations recited in the claim are analogous to those of claim 6 (see Tanguay, Nackman and Schubert as applied to claim 6 above).

With respect to claim 22 (currently amended), the limitations recited in the claim are analogous to those of claim 7 (see Tanguay, Nackman and Schubert as applied to claim 7 above). Note that Tanguay also discloses a machine-readable medium (see, for example, memory 130 in FIG. 1) and a processor (see, for example, CPU 170 in FIG. 1).

With respect to claim 23 (currently amended), the limitations recited in the claim are analogous to those of claim 8 (see Tanguay, Nackman and Schubert as applied to claim 8 above).

With respect to claim 24 (currently amended), the limitations recited in the claim are analogous to those of claim 9 (see Tanguay, Nackman and Schubert as applied to claim 9 above).

With respect to claim 33 (new), the limitations recited in the claim are analogous to those of claim 7 (see Tanguay, Nackman and Schubert as applied to claim 7 above). Note that the recited "IsScan variable" corresponds to the "scan variable" of claim 7.

With respect to claim 34 (new), the limitations recited in the claim are analogous to those of claim 7 (see Tanguay, Nackman and Schubert as applied to claim 7 above). Note that the recited "hidden token type variable" corresponds the "visibility variable" of claim 7.

With respect to claim 35 (new), the limitations recited in the claim are analogous to those of claim 8 (see Tanguay, Nackman and Schubert as applied to claim 9 above).

With respect to claim 36 (new), the limitations recited in the claim are analogous to those of claim 9 (see Tanguay, Nackman and Schubert as applied to claim 9 above).

With respect to claim 37 (new), Schubert further discloses the limitation wherein generating comprises:

(a) receiving a token from a scan insertion module (see, for example, column 13, lines 48-49, which shows receiving the HDL representation, i.e. the tokens);

(b) determining whether the token involves scan related changes (see, for example, column 13, lines 50-65, which shows determining scan related changes, and column 23, lines 43-62, which further shows determining whether portions of the HDL representation, i.e. the tokens, involve scan related changes); and

(c) writing the token to an output file if the token is not scan related (see, for example, column 13, line 66 to column 14, line 7, which shows writing the original HDL representation, i.e. the tokens that are not scan related, to an output file).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the system of Tanguay and Nackman with the features taught by Schubert, so as to debug hardware designs.

With respect to claim 38 (new), Schubert further discloses determining whether more tokens are to be received from the scan insertion module and repeating determining whether the token involves scan related changes and writing the token to an output file until no tokens remain (see, for example, column 13, line 48 to column 14, line 7, which shows performing the steps for the entire HDL representation, i.e. repeating the steps for each token).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the system of Tanguay and Nackman with the features taught by Schubert, so as to debug hardware designs.

With respect to claim 39 (new), the limitations recited in the claim are analogous to those of claim 2 (see Tanguay and Nackman as applied to claim 2 above, in view of Schubert).

With respect to claim 40 (new), the limitations recited in the claim are analogous to those of claim 2 (see Tanguay and Nackman as applied to claim 2 above, in view of Schubert).

With respect to claim 41 (new), Schubert further discloses the limitation wherein generating comprises writing the scan inserted tokens into the HDL file from a buffer that

preserves the text of the original file (see, for example, column 13, line 48 to column 14, line 7, which shows generating a scan inserted HDL file that includes the text of the original file).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the system of Tanguay and Nackman with the features taught by Schubert, so as to debug hardware designs.

With respect to claim 42 (new), the limitations recited in the claim are analogous to those of claim 3 (see Tanguay and Nackman as applied to claim 3 above, in view of Schubert).

With respect to claim 43 (new), the limitations recited in the claim are analogous to those of claim 3 (see Tanguay and Nackman as applied to claim 3 above, in view of Schubert).

With respect to claim 44 (new), the limitations recited in the claim are analogous to those of claim 33 (see Tanguay, Nackman and Schubert as applied to claim 33 above). Note that Tanguay also discloses a machine-readable medium (see, for example, memory 130 in FIG. 1) and a processor (see, for example, CPU 170 in FIG. 1).

With respect to claim 45 (new), the limitations recited in the claim are analogous to those of claim 34 (see Tanguay, Nackman and Schubert as applied to claim 34 above).

With respect to claim 46 (new), the limitations recited in the claim are analogous to those of claim 37 (see Tanguay, Nackman and Schubert as applied to claim 37 above).

With respect to claim 47 (new), the limitations recited in the claim are analogous to those of claim 39 (see Tanguay, Nackman and Schubert as applied to claim 39 above).

With respect to claim 48 (new), the limitations recited in the claim are analogous to those of claim 41 (see Tanguay, Nackman and Schubert as applied to claim 41 above).

With respect to claim 49 (new), the limitations recited in the claim are analogous to those of claim 33 (see Tanguay, Nackman and Schubert as applied to claim 33 above). Note that Schubert further discloses a scan insertion tool (see, for example, instrumentor 326 in FIG. 3).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the system of Tanguay and Nackman with the features taught by Schubert, so as to debug hardware designs.

With respect to claim 50 (new), the limitations recited in the claim are analogous to those of claim 42 (see Tanguay, Nackman and Schubert as applied to claim 42 above).

With respect to claim 51 (new), the limitations recited in the claim are analogous to those of claim 43 (see Tanguay, Nackman and Schubert as applied to claim 43 above).

With respect to claim 52 (new), the limitations recited in the claim are analogous to those of claim 34 (see Tanguay, Nackman and Schubert as applied to claim 34 above).

With respect to claim 53 (new), the limitations recited in the claim are analogous to those of claim 39 (see Tanguay, Nackman and Schubert as applied to claim 39 above).

With respect to claim 54 (new), the limitations recited in the claim are analogous to those of claim 37 (see Tanguay, Nackman and Schubert as applied to claim 37 above).

Conclusion

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MY

Michael J. Yigdall
Examiner
Art Unit 2122

mjy

TUAN DAM
SUPERVISORY PATENT EXAMINER